

PAT-NO: JP409185589A

DOCUMENT-IDENTIFIER: JP 09185589 A

TITLE: INFORMATION PROCESSING SYSTEM AND POWER SAVING
METHOD
FOR THE SYSTEM

PUBN-DATE: July 15, 1997

INVENTOR-INFORMATION:
NAME
KATO, MASAYA

ASSIGNEE-INFORMATION:
NAME COUNTRY
TOSHIBA CORP N/A

APPL-NO: JP08000201

APPL-DATE: January 5, 1996

INT-CL (IPC): G06F015/16, G06F015/16 , G06F001/32 , G06F001/04

ABSTRACT:

PROBLEM TO BE SOLVED: To effectively suppress the power consumption of the whole system by positively generating inactive processors even in a state where the number of tasks is over that of processors.

SOLUTION: In this information processing system, OS previously knows the resource request quantity of the processors 11 and 12 of the processing unit of each task including OS itself to centralize the group of tasks to the specific processor 11 within a range in which the resources of the processors 11 and 12 are not short (a range in which the sum of the request quantity of the

processor resources is not over 100). Thereby, the other processor 12 is brought into an inactive state so that power source supply for the inactive processor is stopped or a clock frequency is lowered. Thereby the power consumption of the whole system is effectively suppressed.

COPYRIGHT: (C)1997,JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-185589

(43) 公開日 平成9年(1997)7月15日

(51) Int.Cl. ⁹	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/16			G 0 6 F 15/16	E
	4 3 0			4 3 0 B
1/32			1/04	3 0 1 C
1/04	3 0 1		1/00	3 3 2 B

審査請求 未請求 請求項の数 5 O L (全 7 頁)

(21) 出願番号 特願平8-201

(22) 出願日 平成8年(1996)1月5日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 加藤 雅也

東京都青梅市末広町2丁目9番地 株式会

社東芝青梅工場内

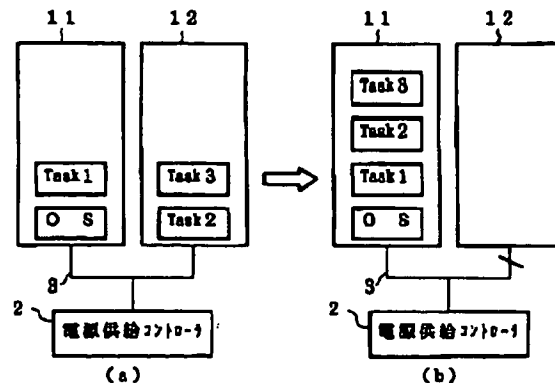
(74) 代理人 弁理士 須山 佐一

(54) 【発明の名称】 情報処理システムと情報処理システムの省電力方法

(57) 【要約】

【課題】 従来、複数のプロセッサを有する情報処理システムにおいては、かりにプロセッサ毎にクロック周波数を可変できたとしても、この省電力機構による消費電力の節減効果は十分には得られないという課題があった。

【解決手段】 本発明の情報処理システムは、OSが、OS自身を含め各タスクの処理単位のプロセッサ資源の要求量を事前に知り、プロセッサ資源が不足しない範囲（プロセッサ資源の要求量の和が100を越えない範囲）でタスク群を特定のプロセッサに集中させる。これにより、他のプロセッサが休止状態となるので、この休止状態のプロセッサに対する電源供給を停止、或いはクロック周波数を下げる。これによりシステム全体の消費電力を効果的に抑制することが可能となる。



【特許請求の範囲】

【請求項1】 複数のタスクを並列に処理可能な複数のプロセッサと、

前記タスクを構成する処理単位毎のプロセッサ資源の要求量に基づき、休止状態のプロセッサが発生するように前記複数のプロセッサのうちの特定のプロセッサにプロセッサ資源の不足が生じない範囲で複数のタスクを集めるタスク管理手段と、

前記休止状態のプロセッサに対する電源供給を停止する電源供給制御手段とを具備することを特徴とする情報処理システム。

【請求項2】 複数のタスクを並列に処理可能な複数のプロセッサと、

前記タスクを構成する処理単位毎のプロセッサ資源の要求量に基づき、休止状態のプロセッサが発生するように前記複数のプロセッサのうちの特定のプロセッサにプロセッサ資源の不足が生じない範囲で複数のタスクを集めるタスク管理手段と、

前記休止状態のプロセッサの動作クロック周波数を下げる手段とを具備することを特徴とする情報処理システム。

【請求項3】 複数のタスクを並列に処理可能な複数のプロセッサを備えた情報処理システムの省電力方法において、

前記タスクを構成する処理単位毎のプロセッサ資源の要求量に基づき、休止状態のプロセッサが発生するように前記複数のプロセッサのうちの特定のプロセッサにプロセッサ資源の不足が生じない範囲で複数のタスクを集め、前記休止状態のプロセッサに対する電源供給を停止することを特徴とする情報処理システムの省電力方法。

【請求項4】 複数のタスクを並列に処理可能な複数のプロセッサを備えた情報処理システムの省電力方法において、

前記タスクを構成する処理単位毎のプロセッサ資源の要求量に基づき、休止状態のプロセッサが発生するように前記複数のプロセッサのうちの特定のプロセッサにプロセッサ資源の不足が生じない範囲で複数のタスクを集め、前記休止状態のプロセッサの動作クロック周波数を下げることを特徴とする情報処理システムの省電力方法。

【請求項5】 請求項3または4記載の情報処理システムの省電力方法において、

前記タスクの処理単位の処理に要した時間と要求処理時間とから前記処理単位のプロセッサ資源の要求量を求めることを特徴とする情報処理システムの省電力方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、複数のプロセッサを備えた情報処理システムとその省電力化の方法に関する。

【0002】

【従来の技術】プロセッサの高速化に伴う消費電力の増大を抑えるための手段としてクロックギアシステムがある。このクロックギアシステムは、プロセッサの動作クロック周波数を段階的に可変できるものとし、プロセッサが休止状態にある時はクロック周波数を下げることによってシステム全体の消費電力の節減効果を得るものである。

【0003】ここで、複数のタスクを並列（時分割多重）に処理することが可能な複数のプロセッサを持つ情報処理システムを考える。また、処理の最大効率を得るため、タスク群は各プロセッサに均等な数で割り当てられるものとする。そして個々のプロセッサのクロック周波数は2段階に可変できるものとする。

【0004】このような情報処理システムにおいて、全てのプロセッサがタスクを処理している場合、全プロセッサのクロック周波数は同一に設定され、全プロセッサの消費電力もほぼ同じである。また、タスクの数がプロセッサの数を下回った場合、休止状態のプロセッサが発生する。そこでこのプロセッサのクロック周波数を下げることによって、このプロセッサの消費電力が節減される。

【0005】以下に、かかる省電力機構を備えた情報処理システムの問題点を述べる。

【0006】上記情報処理システムにおいて消費電力の節減効果が得られるのは、タスクの数がプロセッサの数を下回った場合のような特定の状況に限られる。タスクの数がプロセッサの数以上の場合は、全てのプロセッサがタスク処理を実行するから、全てのプロセッサのクロック周波数は高いほうの値に設定される。タスクの数がプロセッサの数を下回るとは稀であり、よって、かかる従来方式では、消費電力の十分な節減効果が期待できないと考えられる。

【0007】

【発明が解決しようとする課題】このように従来、複数のプロセッサを有する情報処理システムにおいては、かりにプロセッサ毎にクロック周波数を可変できたとしても、この省電力機構による消費電力の節減効果は十分には得られないと言う問題があった。

【0008】本発明はこのような課題を解決するためのもので、タスクの数がプロセッサの数を上回る状況においても休止状態のプロセッサを積極的につくりだすことによって、システム全体の消費電力を効果的に抑制することのできる情報処理システムとその省電力方法の提供を目的としている。

【0009】

【課題を解決するための手段】本発明の情報処理システムは上記した目的を達成するために、複数のタスクを並列に処理可能な複数のプロセッサと、タスクを構成する処理単位毎のプロセッサ資源の要求量に基づき、休止状

態のプロセッサが発生するように複数のプロセッサのうちの特定のプロセッサにプロセッサ資源の不足が生じない範囲で複数のタスクを集めるタスク管理手段と、休止状態のプロセッサに対する電源供給を停止する電源供給制御手段とを具備することを特徴とする。

【0010】また、本発明の情報処理システムの省電力方法は、複数のタスクを並列に処理可能な複数のプロセッサを備えた情報処理システムの省電力方法において、タスクを構成する処理単位毎のプロセッサ資源の要求量に基づき、休止状態のプロセッサが発生するように複数の

10 プロセッサのうちの特定のプロセッサにプロセッサ資源の不足が生じない範囲で複数のタスクを集め、休止状態のプロセッサに対する電源供給を停止することを特徴とする。

【0011】これらの発明においては、タスクを構成する処理単位毎のプロセッサ資源の要求量に基づき、複数のプロセッサのうちの特定のプロセッサにプロセッサ資源の不足が生じない範囲で複数のタスクを集めること

20 で、特定プロセッサをフル稼働に近い状態にする一方、休止状態のプロセッサをつくりだす。この休止状態のプロセッサに対する電源供給を停止することによって、システム全体の消費電力を節減することができる。

【0012】さらに本発明の情報処理システムは上記した目的を達成するために、複数のタスクを並列に処理可能な複数のプロセッサと、タスクを構成する処理単位毎のプロセッサ資源の要求量に基づき、休止状態のプロセッサが発生するように複数のプロセッサのうちの特定の

30 プロセッサにプロセッサ資源の不足が生じない範囲で複数のタスクを集めるタスク管理手段と、休止状態のプロセッサの動作クロック周波数を下げる手段とを具備することを特徴とする。

【0013】また、本発明の情報処理システムの省電力方法は、複数のタスクを並列に処理可能な複数のプロセッサを備えた情報処理システムの省電力方法において、タスクを構成する処理単位毎のプロセッサ資源の要求量に基づき、休止状態のプロセッサが発生するように複数の

40 プロセッサのうちの特定のプロセッサにプロセッサ資源の不足が生じない範囲で複数のタスクを集め、休止状態のプロセッサの動作クロック周波数を下げることを特徴とする。

【0014】これらの発明においては、タスクを構成する処理単位毎のプロセッサ資源の要求量に基づき、複数のプロセッサのうちの特定のプロセッサにプロセッサ資源の不足が生じない範囲で複数のタスクを集めること

【0015】

【発明の実施の形態】以下、この発明の実施の形態につ

いて図面を参照して説明する。

【0016】図1は本実施形態である情報処理システムの全体的な構成を示す図である。

【0017】同図に示すように、この情報処理システムは複数例えば4つのプロセッサ11、12、13、14と各プロセッサへの電源供給を制御する電源供給コントローラ2とを備えて構成される。各プロセッサと電源供給コントローラ2とは相互にバス3及び電源線4を通じて接続されている。個々のプロセッサは各々、複数のタスクを並列（時分割多重）に処理することが可能である。

【0018】この情報処理システムにおいては、タスク群を特定のプロセッサにそのプロセッサの資源が許す範囲で集中させることで、できるだけ多くのプロセッサが休止状態となるように、各プロセッサに対するタスクの割り当てが行われる。

【0019】その様子を図2に示す。簡単化のため、この例では2つのプロセッサ11、12間でのタスクの移動を示す。同図(a)はタスク移動前の状態であり、プロセッサ11はOSとタスク(Task1)を有する。またプロセッサ2はタスク(Task2)とタスク(Task3)を有する。

【0020】ここで、OS、タスクの“プロセッサ資源の要求量”について説明する。“プロセッサ資源の要求量”とは、OS、タスクの処理を行う上でプロセッサの能力のどのくらいの割合を必要とするかを示す値であり、プロセッサをフル稼働させてOS、タスク内の処理単位（関数単位、ブロック単位等の各部位、1つのタスクが1つの処理単位となる場合もある。）を実行した場合の処理時間をT1とし、その処理単位の要求処理時間をT2として、 $(T1/T2) \times 100$ の計算式で求めることができる。例えば、タスク内のある処理単位の要求処理時間T2を10ms、その処理単位をプロセッサをフル稼働させて実行した場合の処理時間T1を5msとすれば“プロセッサ資源の要求量”は $(5/10) \times 100 = 50(\%)$ となる。

【0021】図2において、OSの現在の処理単位のプロセッサ資源の要求量は25、同じくタスク(Task1)は30、タスク(Task2)は20、タスク(Task3)は20とする。また2つのプロセッサ11、12の性能は同一とする。図2(a)のタスク移動前の状態において、一方のプロセッサ11が所有するOSとタスク(Task1)の現在のプロセッサ資源の要求量の和は55、他方のプロセッサ12が所有するタスク(Task2)とタスク(Task3)の現在のプロセッサ資源の要求量の和は40である。したがって、プロセッサ11は $100 - 55 = 45$ のプロセッサ資源を余しており、図2(b)に示すように、プロセッサ12の所有するタスク(Task2)とタスク(Task3)をプロセッサ11に移動させても、プロセッサ11においてOSと全タスクを並列に処理す

ることが可能である。

【0022】本実施形態の情報処理システムは、このようなタスクの移動・割り当て処理をOSの管理の下で実行する。OSはOS自身を含め各タスクの処理単位のプロセッサ資源の要求量を事前に知り、プロセッサ資源が不足しない範囲（プロセッサ資源の要求量の和が100を越えない範囲）でタスク群を特定のプロセッサに集中させるように各プロセッサに対するタスクの割り当てを制御する。各タスクには、プロセッサ資源の要求量を示す情報が処理単位毎に付加されており、これを基にOSは現在の各タスクのプロセッサ資源の要求量を知る。勿論、OS自身にもプロセッサ資源の要求量を示す情報が付加されている。

【0023】図2に示したように、プロセッサ12の全てのタスク（Task2, Task3）をプロセッサ11に移動させたことによって、プロセッサ12は休止状態となる。OSは、プロセッサ12が休止状態になったことを知ると、電源供給コントローラ2に命令を出し、プロセッサ12への電源供給を停止させる。

【0024】図3は新たなタスク（Task4）が発生した場合を示している。このタスク（Task4）のこれから処理しようとしている処理単位のプロセッサ資源の要求量は20である。今、プロセッサ11が所有しているOSと3つのタスク（Task1, Task2, Task3）のプロセッサ資源の要求量の和は95であるから、これに新たなタスク（Task4）のプロセッサ資源の要求量を加えると95+20=115となってプロセッサ11のプロセッサ資源を越えてしまう。そこでこの場合、OSは電源供給コントローラ2に命令を出してプロセッサ12への電源供給を復活させると共に、新たなタスク（Task4）をプロセッサ12に割り当てて、プロセッサ12にタスク（Task4）の処理を実行させる。

【0025】また、図2に示すプロセッサ11がOSと3つのタスク（Task1, Task2, Task3）を保有する状態において、いずれかのタスクのこれから処理しようとする処理単位のプロセッサ資源の要求量が増加してプロセッサ11に対するプロセッサ資源の要求量の和が100を越える場合、OSはこれを判断していずれかのタスクを他方のプロセッサ12に移動させる。

【0026】例えば、図4に示すように、タスク（Task2）のこれから処理する処理単位のプロセッサ資源の要求量が30であるとする。この場合、OSと3つのタスク（Task1, Task2, Task3）のプロセッサ資源の要求量の和は25+30+30+20=105となり、100を越えてしまったため、いずれかのタスクを他方のプロセッサ12に移動させる。この例では、どのタスクを移動させてもプロセッサ11に要求されるプロセッサ資源量の和は100未満となるので、どのタスクをプロセッサ12に移動させてもよい。

【0027】前述したように、以上のタスク移動・割り

当て制御を実現するためには、各タスクにプロセッサ資源の要求量を示す情報を付加しておく必要がある。タスクにプロセッサ資源の要求量を示す情報を付加する方法としては次のようなものを挙げることができる。

【0028】図5において、51はソースプログラム、52はソースプログラム51からオブジェクトプログラム53を生成するコンパイラ、アセンブラ等の言語処理系、54はオブジェクトプログラム53を連結編集して一つのプロセッサ実行形式のプログラム（タスク）55を生成するリンカである。

【0029】タスクにプロセッサ資源の要求量を示す情報を付加する第1の方法は、プログラム製作者自身がタスクの処理単位毎のプロセッサ資源の要求量を求めることによってソースコード或いはオブジェクトコードで記述された資源記述ファイル6を作成し、言語処理時或いはリンカ時に、資源記述ファイル6の記述内容をタスク本体に付加する方法である。また、タスクのソースプログラム1自体にプロセッサ資源の要求量のソースコードを一体化させてもよい。この場合、タスクのソースプログラム1の各部位にC言語におけるpragmaのような記述方法でプロセッサ資源の要求量を示す情報を挿入する方法が考えられる。

【0030】第2の方法は、第1の方法の資源記述ファイル6に代えてタスクの処理単位の要求処理時間を記述した要求処理時間記述ファイルをプログラム製作者自身が作成し、言語処理系が、そのファイルに記述された要求処理時間に基づいて各処理単位のプロセッサ資源の要求量を算出してオブジェクトコード化し、これをタスク本体のオブジェクトコードに付加する方法である。この第2の方法は、第1の方法に比べプログラム製作者の作業負担を軽くすることができる。

【0031】第3の方法は、プロセッサにおいてタスクの処理を実際に実行してみて、その処理に要した時間と、第2の方法で用いた要求処理時間記述ファイルに記述された要求処理時間とからプロセッサ資源の要求量を計算する方法である。前述したように、プロセッサ資源の要求量は、例えば、プロセッサをフル稼働させてタスク内の処理単位を実行した場合の処理時間をT1、その処理単位の要求処理時間をT2として $(T1/T2) \times 100$ の計算式で求めることができる。

【0032】このプロセッサ資源の要求量の計算は、情報処理システムを実際に運用する前にプロファイリング期間を設け、このプロファイリング期間に全てのタスクの処理単位について行うようにすることが望ましい。

【0033】かくして本実施形態の情報処理システムによれば、特定のプロセッサにそのプロセッサ資源の不足が生じない範囲で多くのタスクを集中させることで、その他のプロセッサを積極的に休止状態にし、これら休止状態のプロセッサに対する電源供給を停止することによって、システム全体の消費電力を効果的に抑制すること

が可能となる。

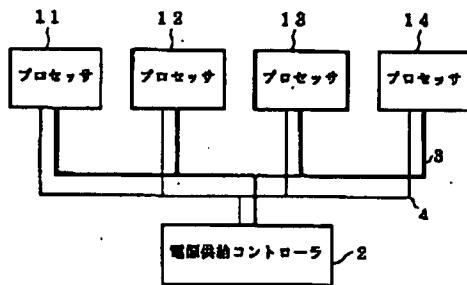
【0034】なお、本実施形態では、休止状態のプロセッサに対する電源供給を停止するようにしたが、プロセッサ毎のクロック周波数を可変できるように構成し、休止状態のプロセッサに対するクロック周波数を下げることによっても消費電力の節減効果が得られる。

【0035】

【発明の効果】以上説明したように本発明によれば、タスクを構成する処理単位毎のプロセッサ資源の要求量に基づき、複数のプロセッサのうちの特定のプロセッサにプロセッサ資源の不足が生じない範囲で複数のタスクを集めることで、特定プロセッサをフル稼働に近い状態にする一方、休止状態のプロセッサをつくりだし、この休止状態のプロセッサに対する電源供給を停止することによって、システム全体の消費電力を節減することができる。

【0036】また、本発明によれば、タスクを構成する処理単位毎のプロセッサ資源の要求量に基づき、複数のプロセッサのうちの特定のプロセッサにプロセッサ資源の不足が生じない範囲で複数のタスクを集めることで、特定プロセッサをフル稼働に近い状態にする一方、休止

【図1】



状態のプロセッサをつくりだし、この休止状態のプロセッサの動作クロック周波数を下げることによって、システム全体の消費電力を節減することができる。

【図面の簡単な説明】

【図1】本実施形態である情報処理システムの全体的な構成を示す図

【図2】図1の情報処理システムにおけるタスク移動を説明するための図

【図3】図1の情報処理システムにおいて新たなタスクが発生した場合のタスク割り当てを説明するための図

【図4】図1の情報処理システムにおいて処理単位のプロセッサ資源の要求量が増大した場合のタスク移動を説明するための図

【図5】プロセッサ資源の要求量の情報をタスクに付加する方法を説明するための図

【符号の説明】

2……電源供給コントローラ

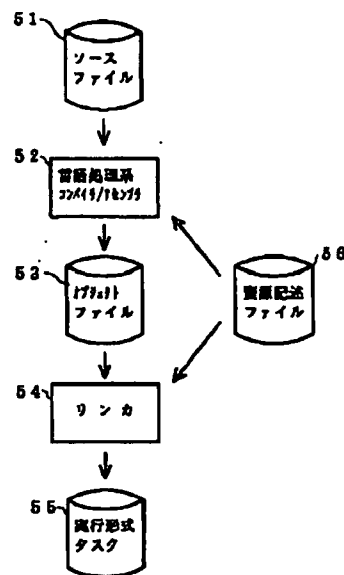
3……バス

4……電源線

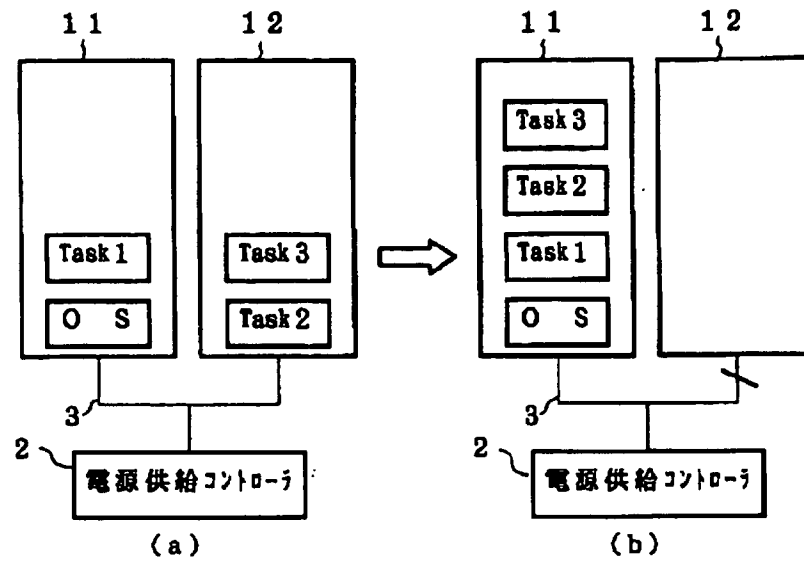
11、12、13、14……プロセッサ

56……資源記述ファイル

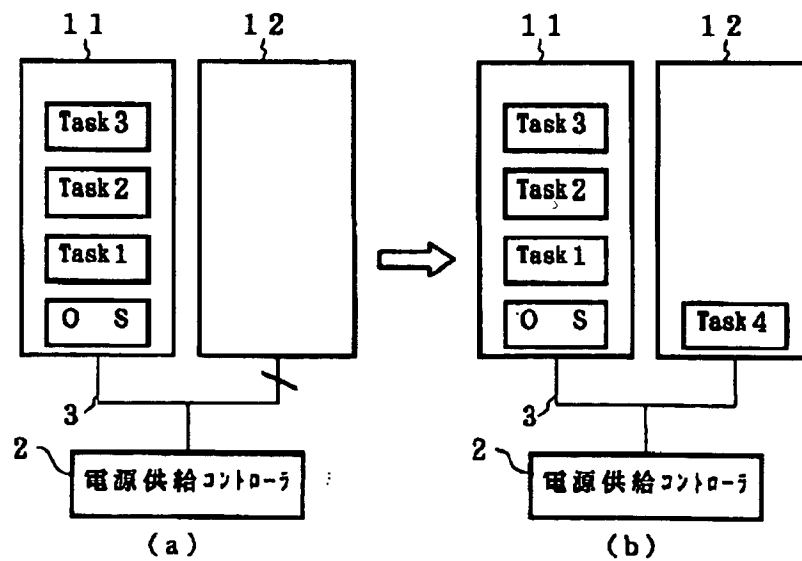
【図5】



【図2】



【図3】



【図4】

